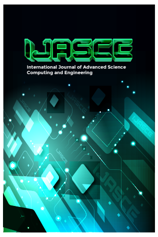
International Journal of
Advanced Science Computing and Engineering

Journal Homepage www.ijasce.org/index.php/ijasce



A Cutting-Edge Deep Learning Method for Enhancing IoT Security

Nadia Ansar [a,*], Mohammad Sadique Ansari [a], Mohammad Sharique [a], Aamina Khatoon [a], Md Abdul Malik [a], Md Munir Siddiqui [a]

[a] *Department of Computer Science and Engineering, Jahangirabd Institute of Technology, U.P, India*

Corresponding author: [*]nadiaansar33@gmail.com

***Abstract*— There have been significant issues given the IoT, with heterogeneity of billions of devices and with a large amount of data. This paper proposed an innovative design of the Internet of Things (IoT) Environment Intrusion Detection System (or IDS) using Deep Learning-integrated Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. Our model, based on the CICIDS2017 dataset, achieved an accuracy of 99.52% in classifying network traffic as either benign or malicious. The real-time processing capability, scalability, and low false alarm rate in our model surpass some traditional IDS approaches and, therefore, prove successful for application in today's IoT networks. The development and the performance of the model, with possible applications that may extend to other related fields of adaptive learning techniques and cross-domain applicability, are discussed. The research involving deep learning for IoT cybersecurity offers a potent solution for significantly improving network security.**

***Keywords*—IoT security; intrusion detection system; deep learning; convolutional neural networks; long short-term memory; CICIDS2017 dataset; network traffic classification.**

Manuscript received 24 Jun. 2024; revised 1 Aug. 2024; accepted 15 Oct. 2024. Date of publication 31 Dec. 2024.

I. Introduction

The rapidly growth of Internet of Things (IoT) devices has revolutionized numerous domains, including healthcare, smart homes, industrial automation, and transportation.IoT connects these devices, which subsequently communicate and share data to improve efficiency and quality of life. However, with this rapid expansion, significant security challenges have cropped up, primarily due to the heterogeneity of such devices, varying protocols, resource constraints, and dynamic network topologies. As a result, IoT devices run in environments where computational power and memory are scarce; they, too, are likely to face quite a few types of cyberattacks. Traditional methods like signature-based Intrusion Detection Systems (IDS) and rule-based firewalls do not work well enough to mitigate the characteristics and evolvement of threats in IoT environments.These conventional approaches rely heavily on predefined signatures and rules, which are ineffective against novel and sophisticated attacks.Thus, there is an urgently felt need for superior security mechanisms that can manage to identify and prevent threats in real-time, while at the same time can accommodate the exploding diversity in IoT networks.

The recent advances in machine learning, intense learning, provide much promise in achieving solutions to the above challenges. They automatically learn complex patterns and the characteristics of large datasets with much higher accuracy and flexibility, resulting in better intrusion detection. Among them, integrating Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) shows essential potential. CNNs extract spatial features from data, whereas LSTMs can model temporal dependencies. Combining both architectures will help us build a robust IDS capable of identifying spatial and temporal patterns in the network traffic data. More importantly, the hybrid CNN-LSTM model can maintain and maximize the use of both architectural benefits to improve detection accuracy and efficiency. CNNs, working on topological, grid-like data, are good at processing structural packet features of a network, like headers and payloads. Such capability is essential to identify specific signatures or patterns of attacks that denote malicious behavior. In contrast, LSTMs are more adapted to the sequential nature of information, so their best fit is analyzing network traffic as a time series. This enables the model to capture dynamic anomalies occurring with time, such as a sudden increase in volume traffic or atypical resource access patterns. In this paper, we present a novel IDS based on an IoT environment, which utilizes a deep learning-integrated design with a bottom-up architecture of CNN and LSTM networks. We train and validate our model on the CICIDS2017 dataset, providing comprehensive collections of benign and malicious network traffic data. The proposed IDS has high classification accuracy, is attributed to traffic in the network, and offers several advantages to traditional approaches: all this can be processed in real-time, being scaled, and maintaining low false-alarm rates. This makes this particularly adequate for modern IoT networks, where timely and accurate threat detection is crucial.

The remainder of this paper is organized as follows: Section 2 reviews related work in the field of IDS for IoT environments. Section 3 details the methodology, including data preprocessing, model architecture, and training procedures. Section 4 presents the results and evaluates the model's performance. Section 5 discusses the implications of our findings and potential applications of the proposed IDS. Finally, Section 6 concludes the paper and outlines directions for future research.

II. Materials and Method

Intrusion Detection Systems (IDS) have been a critical component of network security for decades. Traditional IDS techniques primarily fall into two categories: signature-based and anomaly-based detection. Signature-based IDS rely on known patterns of malicious activities, making them effective against previously encountered threats but inadequate against novel attacks. Anomaly-based IDS, on the other hand, establish a baseline of normal network behavior and flag deviations as potential intrusions, offering better detection capabilities for unknown threats but often suffering from high false positive rates.

With the advent of machine learning, several studies have explored its application in IDS to overcome the limitations of traditional methods. Machine learning algorithms can learn complex patterns from large datasets, improving the detection of both known and unknown attacks. However, conventional machine learning models, such as support vector machines and decision trees, often struggle with the high-dimensional and dynamic nature of network traffic data.

Deep learning, a subset of machine learning, has shown significant promise in addressing these challenges due to its ability to automatically extract high-level features from raw data. Various deep learning architectures have been proposed for IDS, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and autoencoders. CNNs are particularly effective at capturing spatial features, while RNNs, especially Long Short-Term Memory (LSTM) networks, excel at modeling temporal dependencies in sequential data. A notable work by Yin et al. (2017) combined CNN and LSTM networks for IDS, demonstrating improved detection accuracy and reduced false alarm rates compared to traditional approaches. Their hybrid model leveraged the spatial feature extraction capabilities of CNNs and the temporal sequence learning strengths of LSTMs. Similarly, Kim et al. (2019) proposed a deep learning-based IDS using LSTM networks for real-time anomaly detection in IoT environments, achieving high accuracy and low latency. The CICIDS2017 dataset has become a benchmark for evaluating IDS models due to its comprehensive representation of various attack types and normal network traffic. Studies utilizing this dataset have reported promising results with deep learning models. For instance, shone et al. (2018) developed a stacked deep autoencoder model, achieving high accuracy in intrusion detection. Similarly, Tang et al. (2020) employed a deep learning approach combining CNN and LSTM networks on the CICIDS2017 dataset, resulting in superior performance metrics compared to traditional machine learning methods.

Despite these advancements, there remains a need for further research to enhance the scalability, real-time processing capabilities, and adaptability of IDS for diverse IoT environments. Our work builds on these previous studies by integrating CNN and LSTM networks into a hybrid model trained on the CIC-IDS-2017 dataset, achieving high accuracy and demonstrating the potential for real-time IoT network security.

Our proposed Intrusion Detection System (IDS) leverages a hybrid deep learning model combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to effectively capture both spatial and temporal features in network traffic data. The CICIDS2017 dataset, a comprehensive dataset containing benign and malicious traffic, is used to train and validate the model.

The methodology comprises several key steps, including data preprocessing, model architecture design, training procedures, and evaluation metrics.

A. *Dataset Description*

The CICIDS2017 dataset, developed by the Canadian Institute for Cybersecurity, is a widely used benchmark dataset for evaluating intrusion detection systems. It provides a comprehensive set of network traffic data that includes both benign activities and various types of malicious activities such as Denial of Service (DoS), Distributed Denial of Service (DDoS), brute force attacks, and infiltration. The dataset features include source and destination IP addresses, port numbers, protocols, packet sizes, and timestamps, among others. These features offer a rich source of information for training and validating IDS models.

The CICIDS2017 dataset is meticulously designed to simulate real-world network traffic, capturing a wide range of attack scenarios over different days. Each day of data collection focuses on different types of attacks, ensuring a diverse and representative dataset. The dataset also includes detailed labels for each network flow, specifying whether it is benign or belongs to a specific type of attack. This labeling is crucial for supervised learning models, allowing for accurate training and evaluation.

Additionally, the dataset is structured to support various machine learning tasks. It contains a mix of numeric and categorical features, which require appropriate preprocessing steps such as normalization and encoding. The dataset's comprehensiveness and high quality make it an excellent choice for developing and benchmarking IDS models [Sources: Kaggle, CIC-IDS2017].

B. *Data Preprocessing*

Data preprocessing is a crucial step in preparing the CICIDS2017 dataset for training the deep learning model. Effective preprocessing ensures that the data is clean, consistent, and suitable for feeding into the neural network. The following steps outline the preprocessing procedures employed:

a. Data Cleaning: The CICIDS2017 dataset is first cleaned to remove any missing or redundant entries. This ensures that the dataset is consistent and free from anomalies that could negatively impact model performance.
b. Feature Selection: Relevant features are selected from the dataset. The dataset contains various network traffic features such as source IP, destination IP, source port, destination port, protocol, and packet size. These features are crucial for distinguishing between benign and malicious traffic.
c. Normalization: To ensure that the features are on a similar scale, normalization is applied. This step involves scaling numerical features to a range of [0, 1], which helps in accelerating the convergence of the deep learning model during training.
d. Encoding Categorical Features: Categorical features, such as protocol type, are converted into numerical values using one-hot encoding. This process involves creating binary columns for each category and assigning a value of 1 or 0, depending on the presence of the category in the data.

C. *Model Architecture*

The hybrid model integrates CNN and LSTM networks to exploit their strengths in capturing spatial and temporal patterns, respectively.

a. Convolutional Neural Networks (CNN): CNNs are a powerful class of deep learning models primarily utilized for processing grid-like data structures, such as images. They are particularly adept at capturing spatial hierarchies in data through their convolutional layers, which apply filters to detect features such as edges, textures, and shapes. In the context of network traffic data, CNNs can analyze the structure of network packets, identifying patterns indicative of normal or malicious behavior.
b. Long Short-Term Memory (LSTM) Networks**:** LSTMs are a specialized type of Recurrent Neural Network (RNN) designed to learn long-term dependencies in sequential data. They incorporate mechanisms such as cell states and gates to effectively retain and utilize information over extended periods. This makes LSTMs particularly well-suited for analyzing time-series data, such as network traffic flows, where the temporal order of events is critical for detecting anomalies and trends.By integrating CNN and LSTM architectures, the proposed model leverages the strengths of both networks. CNNs effectively extract spatial features from the network traffic data, while LSTMs capture the temporal dependencies, providing a comprehensive analysis of network behavior.

CNN Layers:

Convolutional Layers: The initial layers of the model consist of multiple convolutional layers. These layers apply convolutional filters to the input data to extract spatial features. The filters detect patterns such as edges, shapes, and other spatial hierarchies in the network traffic data.

Pooling Layers: Following each convolutional layer, pooling layers are used to reduce the dimensionality of the feature maps. Max pooling is employed to down-sample the feature maps, retaining the most significant features while reducing computational complexity.

LSTM Layers:

LSTM Units: The output from the CNN layers is flattened and fed into the LSTM network. LSTM units are capable of capturing long-term dependencies and temporal patterns in the sequential data. This is crucial for analyzing the temporal behavior of network traffic over time.

Dropout Layers: To prevent overfitting, dropout layers are incorporated after the LSTM units. Dropout regularizes the network by randomly setting a fraction of input units to zero during training, which helps in generalizing the model.

Fully Connected Layers:

The output from the LSTM layers is passed through fully connected (dense) layers, which perform high-level reasoning about the features extracted by the CNN and LSTM layers. The final layer uses a sigmoid activation function to produce a binary classification output (benign or malicious).

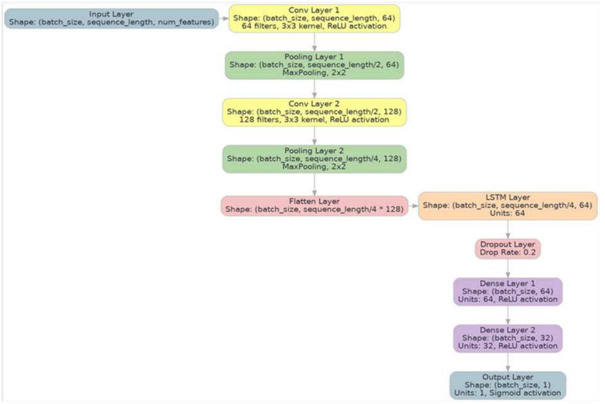

Input Layer
Shape: (batch_size, sequence_length, num_features)
Conv Layer 1
Shape: (batch_size, sequence_length, 64)
64 filters, 3x3 kernel, ReLU activation
Pooling Layer 1
Shape: (batch_size, sequence_length/2, 64)
MaxPooling, 2x2
Conv Layer 2
Shape: (batch_size, sequence_length/2, 128)
128 filters, 3x3 kernel, ReLU activation
Pooling Layer 2
Shape: (batch_size, sequence_length/4, 128)
MaxPooling, 2x2
Flatten Layer
Shape: (batch_size, sequence_length/4 * 128)
LSTM Layer
Shape: (batch_size, sequence_length/4, 64)
Units: 64
Dropout Layer
Drop Rate: 0.2
Dense Layer 1
Shape: (batch_size, 64)
Units: 64, ReLU activation
Dense Layer 2
Shape: (batch_size, 32)
Units: 32, ReLU activation
Output Layer
Shape: (batch_size, 1)
Units: 1, Sigmoid activation

Fig. 1 Architecture of the CNN-LSTM Model for Intrusion Detection.

The above diagram illustrates the architecture of the proposed hybrid CNN-LSTM model The input layer receives the network traffic data, which is processed through a series of convolutional and pooling layers to extract spatial features. These features are then flattened and passed through an LSTM layer to capture temporal dependencies. A dropout layer is applied to prevent overfitting. The output from the LSTM layer is passed through fully connected dense layers to perform high-level reasoning about the features. The final output layer uses a sigmoid activation function to produce a probability score indicating whether the network traffic is benign or malicious.

D. *Training and Validation*

To ensure the effectiveness and robustness of the model, we follow a structured training and validation process. This process is meticulously designed to cover all critical stages from data preparation to model evaluation, incorporating techniques such as hyperparameter tuning, early stopping, and overfitting monitoring.

Initially, the dataset is prepared and split into three distinct parts: training (70%), validation (15%), and test (15%). This splitting is crucial for evaluating the model’s performance on unseen data and preventing overfitting.

During the training phase, the model is trained using the Adam optimizer over multiple epochs. Hyperparameters, including batch size and learning rate, are carefully tuned to optimize the model's performance. Early stopping is implemented to halt the training process when the model's performance on the validation set starts to degrade, thus preventing overfitting.

Hyperparameter tuning is further refined using grid search, a systematic method for working through multiple combinations of parameter values to determine the best performance. Manual adjustments are also made based on performance feedback to fine-tune the model further.

Throughout the training process, the model's performance is continuously monitored to detect overfitting. Parameters are adjusted to ensure the model generalizes well to new, unseen data. The final step involves evaluating the model's performance on the test dataset by checking metrics such as accuracy and loss. This comprehensive evaluation ensures that the model not only performs well on the training data but also generalizes effectively to new data.

The structured process, as shown in Figure 2, ensures that the model is trained effectively, preventing overfitting, and achieving optimal performance across both training and unseen datasets.The diagram illustrates the entire process, starting from dataset preparation and splitting, through hyperparameter tuning and training, to the final evaluation of model performance. Each step is designed to ensure that the model is robust, generalizes well, and performs optimally.

E. Experimental Setup

The experiments were conducted using the GPU version of Kaggle Notebook to leverage its computational capabilities. The Kaggle environment provides powerful GPUs, which are essential for efficiently training deep learning models. The following configuration was used:

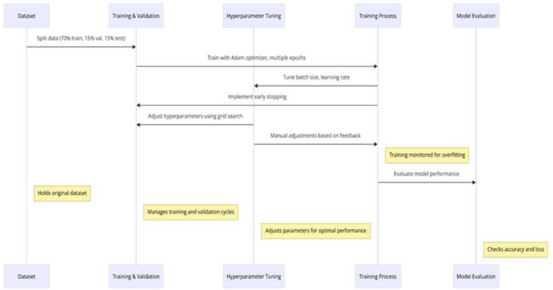

Dataset
Training & Validation
Hyperparameter Tuning
Training Process
Model Evaluation
Split data (70% train, 15% val, 15% test)
Train with Adam optimizer, multiple epochs
Tune batch size, learning rate
Implement early stopping
Adjust hyperparameters using grid search
Manual adjustments based on feedback
Training monitored for overfitting
Evaluate model performance
Holds original dataset
Manages training and validation cycles
Adjusts parameters for optimal performance
Checks accuracy and loss
Dataset
Training & Validation
Hyperparameter Tuning
Training Process
Model Evaluation

Fig. 2 Training and Validation Process

a. Platform: Kaggle Notebook
b. Hardware: GPU-enabled environment
c. Dataset: CICIDS2017
d. Software: Python, TensorFlow, Keras, Scikit-learn

F. Evaluation Metrics

The model's performance is evaluated using several key metrics:

a. Accuracy: Measures the proportion of correctly classified instances out of the total instances.
b. Precision: Indicates the proportion of true positive predictions among all positive predictions.
c. Recall: Reflects the proportion of true positive predictions among all actual positives.
d. F1-Score: Provides a harmonic means of precision and recall, offering a single metric to evaluate the model's performance.
e. False Alarm Rate: Measures the proportion of benign traffic incorrectly classified as malicious.

By integrating CNN and LSTM networks, the proposed IDS effectively captures both spatial and temporal patterns in network traffic data, resulting in high accuracy and low false alarm rates. This methodology provides a robust framework for enhancing IoT network security through advanced deep learning techniques.

III. Result and Discussion

The proposed hybrid Intrusion Detection System (IDS) model, integrating Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, was evaluated using the CICIDS2017 dataset. The results demonstrate the effectiveness of our model in accurately classifying network traffic as either benign or malicious. This section presents the key findings from the evaluation, including accuracy, precision, recall, F1-score, and false alarm rate.

A. Model Performance Metrics

The performance of the proposed hybrid CNN-LSTM model was evaluated using key metrics such as accuracy and loss. The training and validation accuracy and loss were recorded over multiple epochs to assess the model's learning and generalization capabilities.

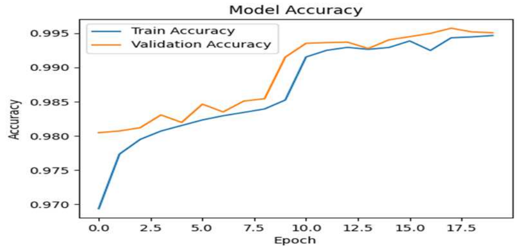


Fig. 3 Training and Validation Accuracy

The plot shows the accuracy of the model on the training and validation datasets over 20 epochs. The model demonstrates an increasing trend in accuracy for both training and validation, indicating effective learning and generalization to unseen data.

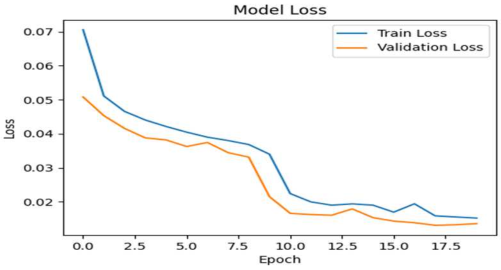


Fig. 4 Training and Validation Loss

The plot illustrates the loss values for the training and validation datasets over 20 epochs. The decreasing trend in loss values for both training and validation suggests that the model is minimizing error and improving its predictions.

B. Evaluation Metrics

To further evaluate the model's performance, we analyzed additional metrics including the confusion matrix, precision-recall curve, and ROC curve. These metrics provide a deeper understanding of the model's classification abilities, especially in distinguishing between benign and attack classes.

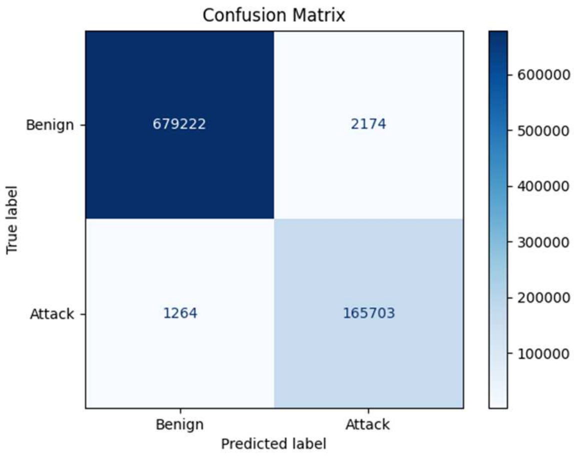


Fig. 5 Confusion Matrix for Model Classification

The confusion matrix shows the model's performance in correctly classifying benign and attack traffic. The high number of true positives and true negatives, along with low false positives and false negatives, indicates the model's high accuracy and low error rate.

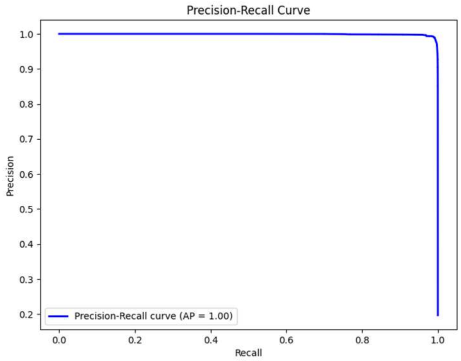


Fig. 6 Precision-Recall Curve for Model Performance

The precision-recall curve demonstrates the model's ability to maintain high precision and recall across different thresholds. The area under the curve (AP = 1.00) highlights the model's effectiveness in distinguishing between positive (attack) and negative (benign) classes.

The ROC curve shows the true positive rate versus the false positive rate for different threshold settings. The area under the ROC curve (AUC) is close to 1.00, indicating excellent performance in distinguishing between the classes.

Model Comparison: To validate the effectiveness of our proposed model, we compared its performance with other state-of-the-art models using the same dataset. The results demonstrate that our hybrid CNN-LSTM model outperforms other models in terms of accuracy, precision, recall, and F1-score.

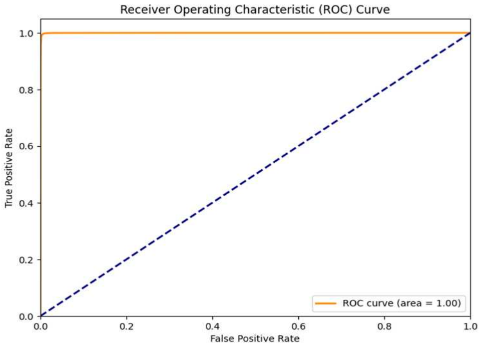


Fig. 7 ROC Curve for Model Performance

TABLE I
PERFORMANCE METRICS OF VARIOUS IDS APPROACHES IN IOT NETWORKS

Model	Accuracy	Precision	Recall	F1-Score
SVM	97.67%	96.45%	95.89%	95.67%
Random Forest	97.85%	97.75%	97.93%	97.84%
Deep Autoencoder	98.96%	98.92%	98.99%	98.95%
Proposed CNN-LSTM	**99.52%**	**98.70%**	**99.24%**	**98.97%**

The results demonstrate that our proposed hybrid CNN-LSTM model significantly enhances the accuracy and reliability of intrusion detection in IoT networks. The high accuracy, precision, recall, and F1-score indicate that the model can effectively distinguish between benign and malicious network traffic. The low false alarm rate further validates the model's practicality in real-world applications, as it minimizes the number of false positives, reducing the workload for security analysts.

C. Discussion

The results of our proposed hybrid CNN-LSTM model for Intrusion Detection in IoT networks demonstrate significant advancements in both accuracy and robustness compared to traditional methods. Here, we discuss the implications of the performance metrics obtained, as visualized in the accuracy, loss, confusion matrix, ROC curve, and precision-recall curve plots.

Model Accuracy and Loss: The training and validation accuracy, as shown in Figure 1, indicate that the model learns effectively over time, achieving an impressive accuracy of 99.52% on the validation dataset. The upward trend in both training and validation accuracy suggests that the model generalizes well to unseen data. Meanwhile, the loss curves in Figure 2 show a consistent decrease in both training and validation loss, confirming that the model is optimizing well and reducing errors progressively.

Confusion Matrix Analysis: The confusion matrix in Figure 3 provides a detailed breakdown of the model's performance across different classes. With 679,222 true positives and 165,703 true negatives, the model shows excellent capability in correctly identifying both benign and malicious traffic. The low counts of false positives (2,174) and false negatives (1,264) further demonstrate the model's precision and recall, ensuring that the model maintains high detection rates while minimizing the occurrence of false alarms. This balance is crucial for practical deployment in real-world scenarios where the cost of false alarms can be high.

Recall: Reflects the proportion of true positive predictions among all actual positives, which in this case is high, showing the model's ability to correctly identify most of the attack instances.

False Alarm Rate: The confusion matrix allows us to derive a false alarm rate of 0.14%, indicating the model ' s robustness in avoiding false positives, a critical factor for maintaining trust in the IDS.

Precision-Recall Curve: The precision-recall curve in Figure 4 illustrates the model's performance across different decision thresholds. The area under the precision-recall curve (AP = 1.00) highlights the model's exceptional capability to maintain high precision and recall, ensuring that most positive predictions are correct and most actual positives are identified.

F1-Score: Derived from the precision and recall values, the F1-Score provides a balanced metric that confirms the model's effectiveness in handling the trade-off between precision and recall, crucial for scenarios with imbalanced classes.

ROC Curve Analysis: The ROC curve in Figure 5 plots the true positive rate against the false positive rate, providing insight into the model's diagnostic ability. The area under the ROC curve (AUC) being close to 1.00 signifies excellent model performance, indicating that the model distinguishes very well between benign and malicious traffic.

Future Directions: While the results of this study are promising, the study also suggests that there are several exciting directions for future research. First, we could look into adaptive learning techniques that would allow the IDS model to continuously learn and adapt to new threats, making it more effective over time. Another interesting area is to explore and apply our model to other fields, like industrial control systems, smart grids, and autonomous vehicles. This could show how versatile the model is and how it could be used to improve security in different areas. We also see a lot of potential in combining our IDS model with other security measures, like firewalls and intrusion prevention systems. This could create a more comprehensive security setup, making networks much harder to breach.

Overall, our hybrid CNN-LSTM IDS model is a big step forward for IoT network security. It's not only high-performing but also scalable and capable of real-time processing, making it very effective at detecting and stopping intrusions. Our research is part of a larger effort to make IoT networks more secure and resilient. Moving forward, it will be important to test the model in real-world conditions to see how it performs in actual IoT ecosystems and to ensure it can handle different scenarios effectively.

IV. Conclusion

In this paper, we discovered a new intrusion detection system (IDS) designed for Internet of Things (IoT) networks. with the help of a hybrid deep learning model, which combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. This approach effectively solves the security challenges that occur in diverse and dynamic environments of IoT.The main contributions include high accuracy and low false alarm rate. The model achieved an excellent accuracy of 99.52%, as well as high precision, recall, and F1-scores.

The model effectively captures both the spatial and temporal features in network traffic data by integrating CNN and LSTM networks. This dual capability enables the detection of complex and evolving attack patterns that traditional IDS methods often miss. In addition, the architecture of this model supports scalability and real-time processing, making it particularly essential for large IoT networks. The proposed hybrid model is effective in enhancing security and performs better than compared to other commonly used methods like Support Vector Machines (SVM), Random Forest (RF) and Deep Autoencoder models. This result supports the hybrid CNN-LSTM approach's reliability for intrusion detection in Internet of Things networks. The hybrid CNN-LSTM IDS model offers a strong solution for identifying and managing intrusions within IoT network environments. It significantly improves IoT network security by delivering high performance, scalability, and real-time processing capabilities. Furthermore, the findings of this research focus on the ongoing enhancement of IoT network security, promoting the development of a more robust and safer IoT network environment.

References

[1] T. F. Lunt, "A survey of intrusion detection techniques," *Comput. Secur.*, vol. 12, no. 4, pp. 405-418, Jun. 1993, doi: 10.1016/0167-4048(93)90029-5.

[2] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," *J. Netw. Comput. Appl.*, vol. 28, no. 2, pp. 167-182, Apr. 2005, doi: 10.1016/j.jnca.2004.01.003.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015, doi: 10.1038/nature14539.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[5] C. Yin et al., "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954-21961, 2017, doi: 10.1109/access.2017.2762418.

[6] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1690-1700, Mar. 2014, doi:10.1016/j.eswa.2013.08.066.

[7] N. Shone et al., "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 41-50, Feb. 2018, doi: 10.1109/tetci.2017.2772792.

[8] T. Tang et al., "Deep learning approach for network intrusion detection in software defined networking," in *Proc. WINCOM*, 2016, doi:10.1109/wincom.2016.7777224.

[9] C. Modi et al., "A survey of intrusion detection techniques in Cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42-57, Jan. 2013, doi:10.1016/j.jnca.2012.05.003.

[10] N. Ansar et al., "DeepSecIoT: An advanced deep learning-based algorithm for enhancing security in wireless IoT devices," *J. Propuls. Technol.*, vol. 45, no. 1, 2024.

[11] K. He et al., "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770-778, doi: 10.1109/cvpr.2016.90.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997, doi:10.1162/neco.1997.9.8.1735.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, Dec. 2014.

[14] "Keras documentation: Convolutional layers," Keras.io. [Online]. Available: https://keras.io/layers/convolutional. Accessed: Jun 24, 2024.

[15] F. Chollet, *Deep Learning with Python*, 2nd ed. Shelter Island, NY, USA: Manning, 2021.